

ОБЩЕЕ ЗАДАНИЕ НА ПРАКТИЧЕСКУЮ ЧАСТЬ И КУРСОВОЕ ПРОЕКТИРОВАНИЕ ПО ДИСЦИПЛИНЕ «АРХИТЕКТУРА ЭВМ»

В рамках выполнения практических заданий и курсового проектирования в процессе изучения дисциплины «Архитектура ЭВМ» необходимо разработать программную модель простейшей вычислительной машины Simple Computer и набор утилит по созданию для неё программного обеспечения.

Архитектура Simple Computer

Архитектура Simple Computer представлена на рисунке 1 и включает следующие функциональные блоки: оперативную память, устройство ввода-вывода (терминал), центральный процессор и генератор тактовых импульсов.

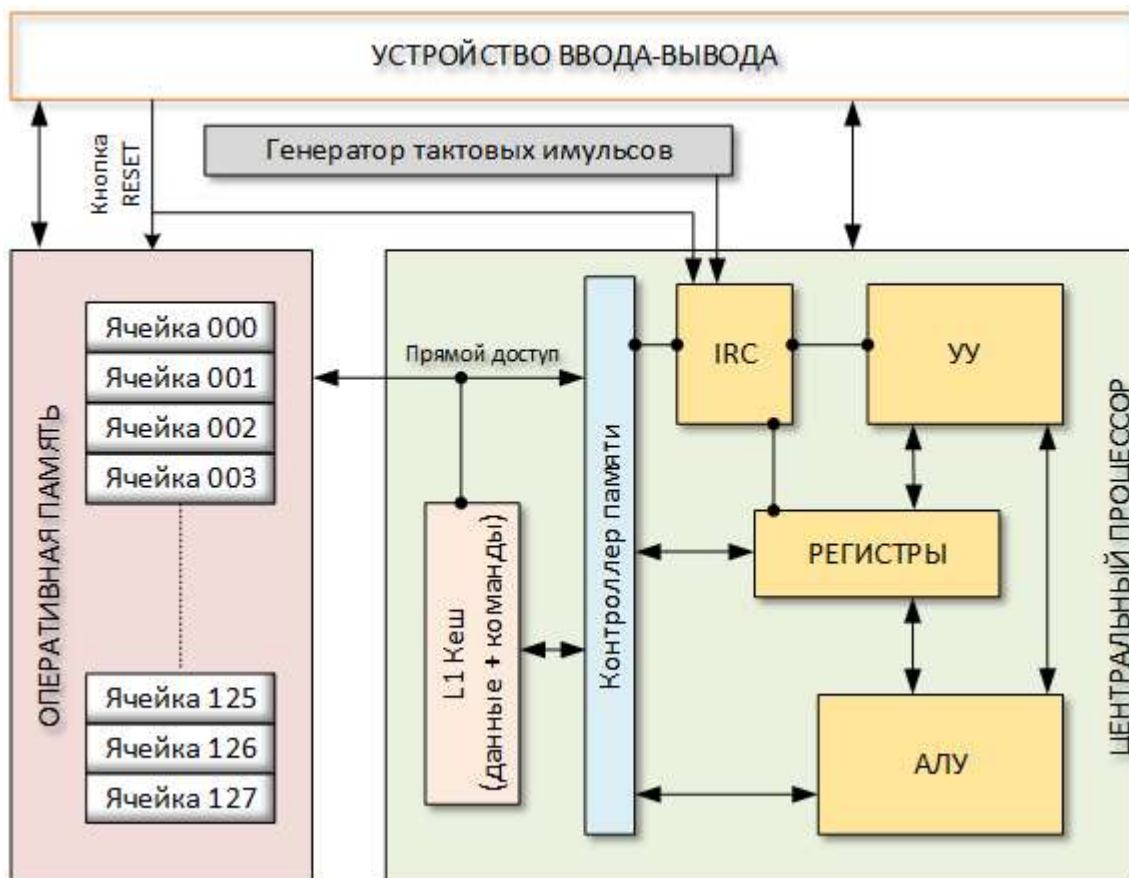


Рисунок 1 – Архитектура вычислительной машины Simple Computer

Блок «Оперативная память»

Оперативная память – это часть Simple Computer, где хранятся программа и данные. **Память состоит из ячеек, каждая из которых хранит 15 двоичных разрядов.**

Ячейка – минимальная единица, к которой можно обращаться при доступе к памяти. Все ячейки последовательно пронумерованы целыми числами. Номер ячейки является её адресом и задается 7-ми разрядным числом. Предполагаем, что Simple Computer **оборудован памятью из 128 ячеек** (с адресами от 0_{10} до 127_{10} или 0_{16} до $7F_{16}$).

Прямой доступ к оперативной памяти имеет центральный процессор (через его контроллер памяти, о нем ниже) и устройство ввода-вывода. Оба эти устройства могут по заданному адресу оперативной памяти (номеру ячейки) поместить значение или считать его.

Блок «Центральный процессор»

Выполнение программ осуществляется центральным процессором Simple Computer. Процессор состоит из следующих функциональных блоков:

- регистры (аккумулятор, счетчик команд, регистр флагов, счетчик тактов простоя процессора);
- арифметико-логическое устройство (АЛУ);
- управляющее устройство (УУ);
- контроллер оперативной памяти;
- кэш L1 данных и команд;
- обработчик прерываний (IRC).

Регистры являются внутренней памятью процессора и используются для хранения данных или отражения состояния процессора. В состав регистров входят:

- **аккумулятор**, используемый для временного хранения данных и результатов операций, имеет разрядность 15 бит;

- **счетчик команд**, указывающий на адрес ячейки памяти, в которой хранится текущая выполняемая команда, разрядность регистра – 15 бит;

- **регистр флагов**, сигнализирующий об определенных событиях. Содержит 5 разрядов, каждый из которых показывает находится ли процессор в соответствующем состоянии (произошло ли соответствующее событие): переполнение при выполнении текущей операции, ошибка деления на 0 при выполнении операции, ошибка выхода за границы памяти, игнорирование тактовых импульсов, получена неверная команда;

- **счетчик тактов простоя процессора** – используется для подсчета тактов, которые должен ждать процессор пока происходит обмен с оперативной памятью, разрядность регистра – 8 бит.

Арифметико-логическое устройство (англ. arithmetic and logic unit, ALU) — блок процессора, который служит для выполнения логических и арифметических преобразований над данными. В качестве данных могут использоваться значения, находящиеся в аккумуляторе, заданные в операнде команды или хранящиеся в оперативной памяти. Результат выполнения операции сохраняется в аккумуляторе или может помещаться сразу в оперативную память.

В ходе выполнения операций АЛУ устанавливает значения флагов «деление на 0» и «переполнение при выполнении текущей операции». Считается, что все операции (кроме доступа к памяти) АЛУ выполняет за один такт работы процессора.

Управляющее устройство (англ. control unit, CU) координирует работу центрального процессора. По сути, именно это устройство отвечает за выполнение программы, записанной в оперативной памяти. Логика работы управляющего устройства следующая: при поступлении сигнала от контроллера прерываний:

1. Если это сигнал Reset, то управляющее устройство задает регистрам процессора значения по умолчанию и обнуляет всю оперативную память. По умолчанию регистры процессора Simple Computer имеют следующие значения:
 - a. Аккумулятор = 0;
 - b. Счетчик команд = 0;
 - c. Счетчик тактов простоя процессора = 0;
 - d. Флаг «переполнение при выполнении операции» = 0;
 - e. Флаг «ошибка деления на 0» = 0;
 - f. Флаг «Ошибка выхода за границы памяти» = 0;
 - g. Флаг «Получена неверная команда» = 0;
 - h. Флаг «Игнорирование тактовых импульсов» = 1.
2. Если пришел сигнал от генератора тактовых импульсов, то управляющее устройство:
 - a. получает текущую выполняемую команду из памяти (адрес текущей команды указан в регистре счетчика команд IC);
 - b. декодирует её. Если в процессе декодирования произошла ошибка, то устанавливает флаги «указана неверная команда» и «игнорирование тактовых импульсов» и работа управляющего устройства завершается;
 - c. в зависимости от полученной команды:

- i. если это арифметическая или логическая операция, то команда и операнд передается в АЛУ и ожидается обработка команды;
- ii. если это команда управления, то соответствующим образом меняется состояние регистров процессора;
- iii. если это команда взаимодействия с устройством ввода/вывода, то команда и операнд передается устройству ввода/вывода и ожидается выполнение соответствующей операции;
- d. определяется следующая выполняемая команда (меняется состояние регистра счетчика команд, если это необходимо в соответствии с текущей обрабатываемой командой). По умолчанию считается, что ячейки памяти обрабатываются последовательно. Исключение – команды управления, которые задают в своем операнде номер ячейки памяти следующей команды.

Блок «Генератор тактовых импульсов», кнопка Reset и контроллер прерываний

Генератор тактовых импульсов используется для синхронизации действий элементов Simple Computer. **Импульсы генерируются с частотой 2 импульса в секунду** и направляются в центральный процессор, а точнее в его модуль – контроллер прерываний.

Получив очередной импульс контроллер прерываний проверяет состояние флага «Игнорирование тактовых импульсов». Если флаг установлен, то импульс игнорируется и на этом обработка прерывания останавливается.

Далее проверяется содержимое счетчика тактовых импульсов простоя процессора. Если это значение больше 0, то оно уменьшается на 1 и обработка прерывания завершается. Если значение этого счетчика равно 0, то сигнал передается блоку управляющего устройства для выполнения очередной команды.

Если приходит прерывание от нажатия кнопки Reset, то этот сигнал безусловно передается управляющему устройству для соответствующей реакции на него.

Блок «Контроллер памяти»

Контроллер памяти предназначен для организации доступа блоков процессора к оперативной памяти с целью извлечения оттуда или сохранения туда данных (команд). Оперативная память работает значительно медленнее, чем центральный процессор и **прямой доступ центрального процессора к оперативной памяти занимает 10 тактов**.

С целью оптимизации доступа в центральном процессоре используется **кэш, в котором хранится 5 строк по 10 значений**. Прежде, чем делать запрос к оперативной памяти контроллер памяти проверяет наличие запрашиваемых данных (команд) в кэше. Если они там есть, то запрос выполняется за один такт и значение сразу передается контроллером памяти блоку, который их запрашивал. Если запрашиваемой ячейки нет в кэше, то контроллер запрашивает из оперативной памяти строку (10 значений, выровненных по границе 10) и помещает её в кэш. При этом, если в кэше нет свободных строк, то самая неиспользуемая строка (доступ к ячейкам которой не было дольше всего) выгружается обратно в оперативную память (если она изменялась) и освобождается, а на её место загружается новая строка из оперативной памяти. Для упрощения ситуации считаем, что операция записи выгружаемых данных в оперативную память и чтение новых данных из оперативной памяти осуществляется одним запросом контроллера памяти.

При записи значений в оперативную память данные сохраняются в кэше. При этом если соответствующей строки в кэше не было, то она предварительно загружается в него по аналогии с загрузкой данных при запросе из памяти с учетом простоя процессора.

Если контроллеру памяти передан некорректный адрес ячейки памяти, то устанавливается флаг выхода за границы памяти и работа контроллера прекращается.

Блок «Устройство ввода-вывода»

Блок «Устройство ввода-вывода» используется для организации взаимодействия с пользователем Simple Computer и для управления самой моделью. Интерфейс устройства ввода вывода представлен на рисунке 2.

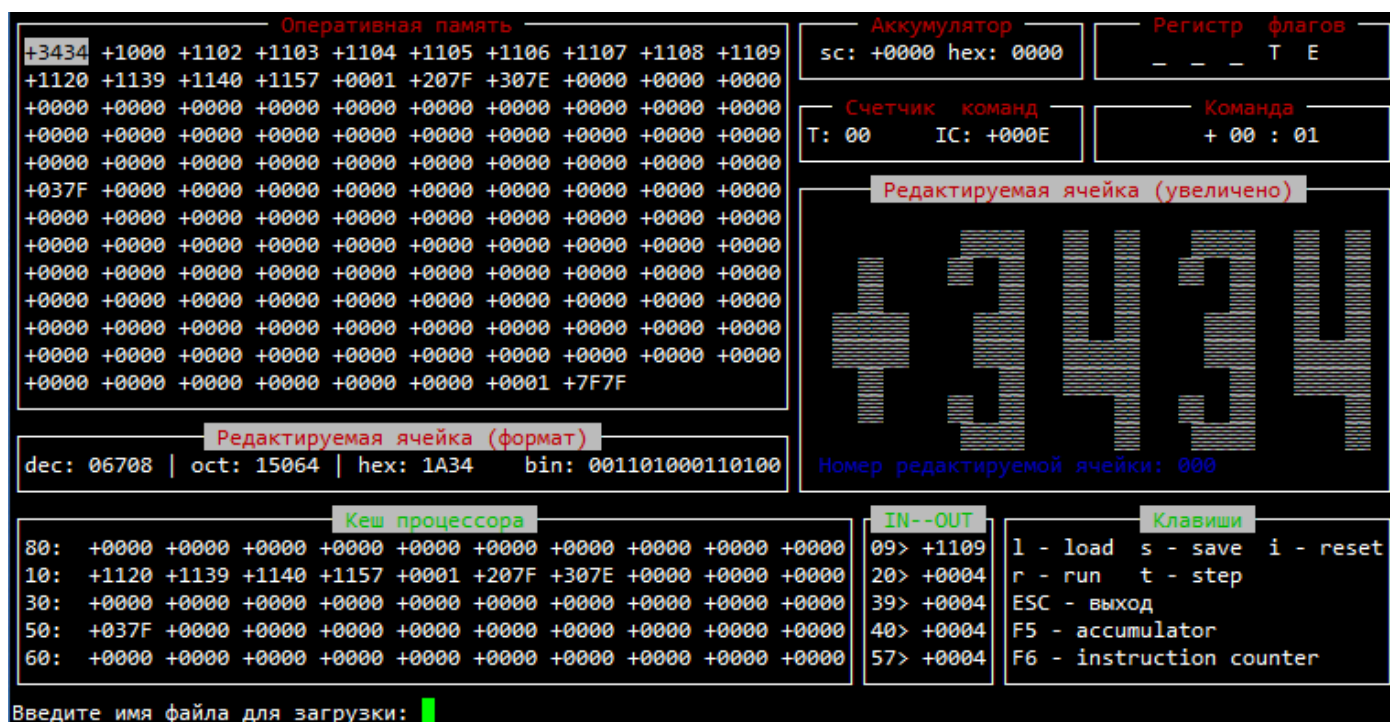


Рисунок 2 – интерфейс консоли управления моделью Simple Computer

Интерфейс разделен на несколько областей:

- “Оперативная память” – содержимое оперативной памяти Simple Computer. В этой области реализован механизм интерактивного редактирования содержимого ячеек памяти. Редактирование доступно только в том случае, если центральный процессор игнорирует тактовые импульсы. Содержимое ячейки памяти выводится и вводится в декодированном виде в соответствии с форматом команд Simple Computer. При выходе за границы блока реализован циклический переход по строкам и столбцам. Редактирование ячейки памяти осуществляется после нажатия на клавишу Enter в том же месте, где выводится значение ячейки. Пользователь вводит значение ячейки в том же виде, как оно выводится на экран (в декодированном), а после того, как пользователь ввел значение оно обратно кодируется и помещается в оперативную память. Во время ввода значение пользователю разрешено вводить только допустимые символы для соответствующего места редактирования (в первом знакоместе только значения + и -, во втором цифры 0-9 и буквы A-E, в третьем цифры 0-9 и буквы A-F и т.п.). Редактирование отменяется при нажатии на клавишу ESC. Завершается нажатием клавиши ENTER;
- «Редактируемая ячейка (формат)», «Редактируемая ячейка (увеличено)» - в этих блоках выводится содержимое текущей ячейки памяти (которая «подсвечена» курсором в блоке «Оперативная память»). В блоке «увеличено» выводится адрес редактируемой ячейки и её значение в увеличенном формате. Значение выводится также в декодированном виде в соответствии с форматом команд Simple Computer. В блоке «формат» не декодированное значение редактируемой ячейки памяти выводится в десятичной системе счисления, в восьмеричной системе счисления, в шестнадцатеричной системе счисления и в двоичной системе счисления.
- Блок вывода состояния регистров:
 - “Аккумулятор” – выводится значение, находящееся в аккумуляторе в декодированном виде и в виде простого целого числа в шестнадцатеричной системе счисления;
 - “Регистр флагов” – выводятся значения флагов. При этом если флаг не установлен (содержит 0), то выводится символ «_», если флаг установлен, то выводится соответствующая буква: P – переполнение ячейки, “0”(ноль) – ошибка деления на 0,

- М - ошибка выхода за границы памяти, Т – игнорирование тактовых импульсов, Е - Получена неверная команда;
- «Счетчик команд» - выводится содержимое регистров «Счетчик команд» (шестнадцатеричный формат) и «Счетчик тактов простоя процессора» (десятичный формат)
 - «Кэш процессора» - выводится содержимое L1 кэша команд и данных. Каждая строка выводится в формате «адрес: строка кэша», где «адрес» — это адрес начала строки кэша, а «строка кэша» — это 10 ячеек памяти, которые сохранены в этой строке. Если строка кэша свободна, то в поле адреса выводится -1 и далее пустая строка.
 - «IN—OUT» — это блок выполнения интерактивного ввода вывода, осуществляемому по указанию центрального процессора. Формат каждой строки – «адрес»«тип» «значение», где «адрес» - это адрес ячейки памяти, с которой проводится взаимодействие, «тип» - это условное отображение вида операции: “>” – вывод значения ячейки памяти, “<» - ввод значения ячейки памяти. Содержимое ячейки памяти выводится и вводится в декодированном виде в соответствии с форматом команд SimpleComputer. Ввод значения осуществляется аналогично вводу значения в блоке «Оперативная память». В блоке реализована прокрутка текста (очередная строка снизу поднимает строки вверх, самая верхняя строка удаляется);
 - «Команда» - выводится поля команды (см. описание системы команд ниже), расположенной в ячейке памяти, на которую ссылается счетчик команд. Если счетчик команд указан некорректно (будет выход за границы памяти), то в поле «Команда» выводится знак “!” и дефолтное значение “+ FF : FF”;
 - «Клавиши» - информационный блок, содержащий подсказку по назначению функциональных клавиш устройства ввода/вывода. Пользователь имеет возможность с помощью клавиш управления курсора выбирать ячейки оперативной памяти и задавать им значения. Нажав клавишу “F5”, пользователь может задать значение аккумулятору, “F6” – регистру «счетчик команд». Сохранить содержимое памяти (в бинарном виде) в файл или загрузить его обратно пользователь может, нажав на клавиши «l», «s» соответственно (после нажатия внизу окна (в информационном поле) устройства ввода/вывода пользователю предлагается ввести имя файла). Запустить программу на выполнение (установить значение флага «игнорировать такты таймера» в 0) можно с помощью клавиши “r”. В процессе выполнения программы, редактирование памяти и изменение значений регистров недоступно. Чтобы выполнить только текущую команду пользователь может нажать клавишу “t”. Обнулить содержимое памяти и задать регистрам значения «по умолчанию» можно нажав на клавишу “i”. Редактирование значения аккумулятора и счетчика команд осуществляется аналогично вводу значения в блоке «Оперативная память»;

Система команд Simple Computer

Формат команды в Simple Computer следующий (см. рисунок 3): старший разряд содержит знак или признак команды (0 – команда или знак +, 1 – знак минус), разряды с 8 по 14 определяют код операции, младшие 7 разрядов содержат операнд. Коды операций, их назначение и обозначение в Simple Assembler и приведены в таблице 1.



Рисунок 3 – Формат команды центрального процессора Simple Computer

Таблица 1. Команды центрального процессора Simple Computer

Операция			Значение
Обозначение	Код		
Операции ввода/вывода			
NOP	00	0x00	Пустая операция. Не выполняется никаких действий
CPUINFO	01	0x01	Требование терминалу вывести в информационном поле вывести информацию об авторе модели Simple Computer в формате «Фамилия Имя Отчество, группа»
Операции ввода/вывода			
READ	10	0x0A	Ввод с терминала в указанную ячейку памяти с контролем переполнения
WRITE	11	0x0B	Вывод на терминал значение указанной ячейки памяти
Операции загрузки/выгрузки в аккумулятор			
LOAD	20	0x14	Загрузка в аккумулятор значения из указанного адреса памяти
STORE	21	0x15	Выгружает значение из аккумулятора по указанному адресу памяти
Арифметические операции			
ADD	30	0x1E	Выполняет сложение слова в аккумуляторе и слова из указанной ячейки памяти (результат в аккумуляторе)
SUB	31	0x1F	Вычитает из слова в аккумуляторе слово из указанной ячейки памяти (результат в аккумуляторе)
DIVIDE	32	0x20	Выполняет деление слова в аккумуляторе на слово из указанной ячейки памяти (результат в аккумуляторе)
MUL	33	0x21	Вычисляет произведение слова в аккумуляторе на слово из указанной ячейки памяти (результат в аккумуляторе)
Операции передачи управления			
JUMP	40	0x28	Переход к указанному адресу памяти
JNEG	41	0x29	Переход к указанному адресу памяти, если в аккумуляторе находится отрицательное число
JZ	42	0x2A	Переход к указанному адресу памяти, если в аккумуляторе находится ноль
HALT	43	0x2B	Останов, выполняется при завершении работы программы
Пользовательские функции			
NOT	51	0x33	Двоичная инверсия слова в аккумуляторе и занесение результата в указанную ячейку памяти
AND	52	0x34	Логическая операция И между содержимым аккумулятора и словом по указанному адресу (результат в аккумуляторе)
OR	53	0x35	Логическая операция ИЛИ между содержимым аккумулятора и словом по указанному адресу (результат в аккумуляторе)
XOR	54	0x36	Логическая операция исключающее ИЛИ между содержимым аккумулятора и словом по указанному адресу (результат в аккумуляторе)
JNS	55	0x37	Переход к указанному адресу памяти, если в аккумуляторе находится положительное число
JC	56	0x38	Переход к указанному адресу памяти, если при сложении произошло переполнение
JNC	57	0x39	Переход к указанному адресу памяти, если при сложении не произошло переполнение
JP	58	0x3A	Переход к указанному адресу памяти, если результат предыдущей операции четный
JNP	59	0x3B	Переход к указанному адресу памяти, если результат предыдущей операции нечетный
CHL	60	0x3C	Логический двоичный сдвиг содержимого указанной ячейки памяти влево (результат в аккумуляторе)
SHR	61	0x3D	Логический двоичный сдвиг содержимого указанной ячейки памяти вправо (результат в аккумуляторе)
RCL	62	0x3E	Циклический двоичный сдвиг содержимого указанной ячейки памяти влево (результат в аккумуляторе)
RCR	63	0x3F	Циклический двоичный сдвиг содержимого указанной ячейки памяти вправо (результат в аккумуляторе)
NEG	64	0x40	Получение дополнительного кода содержимого указанной ячейки памяти (результат в аккумуляторе)
ADDC	65	0x41	Сложение содержимого указанной ячейки памяти с ячейкой памяти, адрес которой находится в аккумуляторе (результат в аккумуляторе)
SUBC	66	0x42	Вычитание из содержимого указанной ячейки памяти содержимого ячейки памяти, адрес которой находится в аккумуляторе (результат в аккумуляторе)
LOGLC	67	0x43	Логический двоичный сдвиг содержимого указанного участка памяти влево на количество разрядов, указанное в аккумуляторе (результат в аккумуляторе)

LOGRC	68	0x44	Логический двоичный сдвиг содержимого указанного участка памяти вправо на количество разрядов, указанное в аккумуляторе (результат в аккумуляторе)
RCCL	69	0x45	Циклический двоичный сдвиг содержимого указанного участка памяти влево на количество разрядов, указанное в аккумуляторе (результат в аккумуляторе)
RCCR	70	0x46	Циклический двоичный сдвиг содержимого указанного участка памяти вправо на количество разрядов, указанное в аккумуляторе (результат в аккумуляторе)
MOVA	71	0x47	Перемещение содержимого указанной ячейки памяти в ячейку, адрес которой указан в аккумуляторе
MOVR	72	0x48	Перемещение содержимого ячейки памяти, адрес которой содержится в аккумуляторе в указанную ячейку памяти.
MOVCA	73	0x49	Перемещение содержимого указанной ячейки памяти в ячейку памяти, адрес которой находится в ячейке памяти, на которую указывает значение аккумулятора
MOVCR	74	0x4A	Перемещение в указанный участок памяти содержимого участка памяти, адрес которого находится в участке памяти, указанном в аккумуляторе
ADDC	75	0x4B	Сложение содержимого указанной ячейки памяти с ячейкой памяти, адрес которой находится в ячейке памяти, указанной в аккумуляторе (результат в аккумуляторе)
SUBC	76	0x4C	Вычитание из содержимого указанной ячейки памяти содержимого ячейки памяти, адрес которой находится в ячейке памяти, указанной в аккумуляторе (результат в аккумуляторе)

Транслятор с языка Simple Assembler

Разработка программ для Simple Computer может осуществляться с использованием низкоуровневого языка Simple Assembler. Для того чтобы программа могла быть обработана Simple Computer необходимо реализовать транслятор, переводящий текст Simple Assembler в бинарный формат, которым может быть считан консолью управления. Пример программы на Simple Assembler:

```

00 READ 09 ; (Ввод A)
01 READ 10 ; (Ввод B)
02 LOAD 09 ; (Загрузка A в аккумулятор)
03 SUB 10 ; (Отнять B)
04 JNEG 07 ; (Переход на 07, если отрицательное)
05 WRITE 09 ; (Вывод A)
06 HALT 00 ; (Останов)
07 WRITE 10 ; (Вывод B)
08 HALT 00 ; (Останов)
09 = +0000 ; (Переменная A)
10 = +9999 ; (Переменная B)

```

Программа транслируется по строкам, задающим значение одной ячейки памяти. Каждая строка состоит как минимум из трех полей: адрес ячейки памяти, команда (символьное обозначение), операнд. Четвертым полем может быть указан комментарий, который обязательно должен начинаться с символа точка с запятой. Название команд представлено в таблице 1. Дополнительно используется команда =, которая явно задает значение ячейки памяти в формате вывода его на экран консоли (+XXXX).

Команда запуска транслятора должна иметь вид: sat файл.sa файл.о, где файл.sa – имя файла, в котором содержится программа на Simple Assembler, файл.о – результат трансляции.

Транслятор с языка Simple Basic

Для упрощения программирования пользователю модели Simple Computer должен быть предоставлен транслятор с высокоуровневого языка Simple Basic. Файл, содержащий программу на Simple Basic, преобразуется в файл с кодом Simple Assembler. Затем Simple Assembler-файл транслируется в бинарный формат.

В языке Simple Basic используются следующие операторы: rem, input, output, goto, if, let, end. Пример программы на Simple Basic:

```

10 REM Это комментарий
20 INPUT A
30 INPUT B
40 LET C = A - B
50 IF C < 0 GOTO 20

```



```
60 PRINT C
70 END
```

Каждая строка программы состоит из номера строки, оператора Simple Basic и параметров. Номера строк должны следовать в возрастающем порядке. Все команды за исключением команды конца программы могут встречаться в программе многократно. Simple Basic должен оперировать с целыми выражениями, включающими операции +, -, *, и /. Приоритет операций аналогичен C. Для того чтобы изменить порядок вычисления, можно использовать скобки.

Транслятор должен распознавать только букв верхнего регистра, то есть все символы в программе на Simple Basic должны быть набраны в верхнем регистре (символ нижнего регистра приведет к ошибке). Имя переменной может состоять только из одной буквы. Simple Basic оперирует только с целыми значениями переменных, в нем отсутствует объявление переменных, а упоминание переменной автоматически вызывает её объявление и присваивает ей нулевое значение. Синтаксис языка не позволяет выполнять операций со строками.